



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA PODNIKATELSKÁ**

FACULTY OF BUSINESS AND MANAGEMENT

**ÚSTAV INFORMATIKY**

INSTITUTE OF INFORMATICS

**MATEMATICKÉ METODY V EKONOMII**

MATHEMATICAL METHODS IN ECONOMICS

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

Ladislav Blaha

**VEDOUCÍ PRÁCE**

SUPERVISOR

Mgr. Veronika Novotná, Ph.D.

**BRNO 2016**

# **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

**Blaha Ladislav**

---

Manažerská informatika (6209R021)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává bakalářskou práci s názvem:

**Matematické metody v ekonomii**

v anglickém jazyce:

**Mathematical Methods in Economics**

Pokyny pro vypracování:

Úvod

Cíle práce, metody a postupy zpracování

Teoretická východiska práce

Analýza současného stavu

Vlastní návrhy řešení

Závěr

Seznam použité literatury

Přílohy

Seznam odborné literatury:

HINDLS, R. Statistika pro ekonomy. 8. vyd. Praha: Professional Publishing, 2007. 415 s. ISBN 978-80-86946-43-6.

KROPÁČ, J. Statistika B. 2. dopl. vyd. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2009. 151 s. ISBN 978-80-214-3295-6.

KUBANOVÁ, J. Statistické metody pro ekonomickou a technickou praxi. 3. vyd. Bratislava: STATIS, 2008. 247 s. ISBN 978-80-85659-474.

RŮČKOVÁ, P. Finanční analýza: metody, ukazatele, využití v praxi. 3. rozš. vyd. Praha: Grada, 2010. 139 s. ISBN 978-80-247-3308-1.

SEDLÁČEK, J. Finanční analýza podniku. 1. vyd. Brno: Computer Press, 2007. 154 s. ISBN 978-80-251-1830-6.

Vedoucí bakalářské práce: Mgr. Veronika Novotná, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2015/2016.

L.S.

---

doc. RNDr. Bedřich Půža, CSc.  
Ředitel ústavu

---

doc. Ing. et Ing. Stanislav Škapa, Ph.D.  
Děkan fakulty

V Brně, dne 29.2.2016

## **Abstrakt**

Tato práce se zabývá analýzou současného stavu veřejné správy a podnikatelského prostředí v Moldavsku, zejména chybějící odbornou terminologií, která je nutná k efektivnímu fungování obou sektorů. Jejím hlavním cílem je vytvoření aplikace, sloužící ke zjištění Levenshteinovi vzdálenosti dvou textových řetězců. Tato aplikace primárně slouží k porovnávání textových řetězců v různých jazycích, jinými slovy zjišťuje míru jejich podobnosti. Tato aplikace bude později využita při tvorbě slovníku, který bude zaměřen na terminologii z oblasti veřejné správy.

## **Abstract**

This bachelor thesis is follow up to an analyses made of actual state of affairs in public administration and business ambience in Moldova, focusing on the missing professional terminology, which is necessary for the effective function of both sectors. The main purpose is to make an application, which would be used to determine the Levenshtein distance from two textual chains. This application is used primarily for comparing textual chains in different languages. In other words it finds the scale of their similarity the application will be used later to create a dictionary, which will focus on terminology from the area of public administration.

## **Klíčová slova**

Moldavsko, Veřejná správa, Levenshteinova vzdálenost, Visual Basic for Applications

## **Key words**

Moldova, Public administration, Levenshtein distance, Visual Basic for Applications

## **Bibliografická citace**

BLAHA, L. *Matematické metody v ekonomii*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2016. 53 s. Vedoucí bakalářské práce Mgr. Veronika Novotná, Ph.D..

## **Čestné prohlášení**

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně.  
Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil  
autorská práva (ve smyslu zákona č. 121/2000Sb., o právu autorském a o právech  
souvisejících s právem autorským).

V Brně .....

podpis .....

## **Poděkování**

Rád bych poděkoval své vedoucí Mgr. Veronice Novotné, Ph.D. za cenné odborné rady a doporučení, které mi velice pomohly při vypracovávání bakalářské práce. Rovněž bych zde rád poděkoval společnosti DCB Actuaries and Consultants s.r.o., za poskytnutí důležitých materiálů a doporučení.

# OBSAH

ÚVOD .....	10
1 CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ .....	11
2 TEORETICKÁ VÝCHODISKA PRÁCE .....	12
2.1 Veřejná správa.....	12
2.1.1 Činnosti veřejné správy .....	13
2.1.2 Funkce veřejné správy .....	14
2.1.3 eGovernment.....	15
2.2 Formální jazyky .....	17
2.2.1 Abeceda .....	17
2.2.2 Řetězec.....	18
2.2.3 Gramatika.....	18
2.3 Metody porovnávání slov.....	19
2.3.1 Levenshteinova vzdálenost .....	19
2.3.2 Damerau-Levenshteinova vzdálenost .....	21
2.3.3 Hammingova vzdálenost.....	22
2.4 Visual Basic for Application (VBA).....	23
2.4.1 Vývojové prostředí Microsoft Visual Basic for Applications .....	23
3 ANALÝZA SOUČASNÉHO STAVU .....	26
3.1 Analýza současného stavu Moldavska.....	26
3.1.1 Místní veřejná správa.....	26
3.1.2 Problémy ve veřejné správě .....	27
3.1.3 Kompetence .....	28
3.1.4 Financování.....	29



3.1.5	Problémy s jazykem.....	29
3.2	DCB Actuaries and Consultants s.r.o.....	30
3.3	Dostupnost aplikace .....	31
4	VLASTNÍ NÁVRH ŘEŠENÍ .....	32
4.1	Naprogramování funkce.....	32
4.2	Aplikování funkce na slovníky .....	36
4.3	Uživatelské rozhraní.....	37
4.3.1	Přidání nového záznamu do slovníku .....	37
4.3.2	Ostatní doplňky u slovníků .....	39
4.3.3	Přidání nových listů do sešitu .....	41
4.3.4	Úvodní list.....	42
4.3.5	Doplňky do Úvodního listu.....	45
4.3.6	List Historie .....	47
4.3.7	Vlastnosti aktivních prvků .....	48
	ZÁVĚR .....	50
	SEZNAM POUŽITÉ LITERATURY .....	51
	SEZNAM PŘÍLOH.....	53

## ÚVOD

Společenství demokratických evropských států, které se zavázaly ke spolupráci v zájmu zachování míru a prosperity, to je Evropská unie. Tato spolupráce není určena pouze na podporu státům, které jsou členy unie, ale i ostatním evropským státům. Aby takové společenství fungovalo, snaží se EU vytvořit jednotné administrativní prostředí. Evropskou unii v současnosti tvoří 28 států a několik dalších se chce připojit. Mezi zeměmi, které se chtějí připojit k EU, se vyskytuje i Moldavsko, kterým se bude má práce zabývat.

Vývoj podnikání v Moldavsku je nutné vnímat v historickém kontextu. Je to z důvodu, že podnikání obecně bylo dlouhou dobu z hospodářského života vyloučeno. Až do 80. let 20. století fungoval v Moldavsku pouze administrativní systém řízení, díky kterému bylo podnikání vytlačeno z ekonomického sektoru a tak se stalo součástí černého trhu. Po vyhlášení nezávislosti 27. srpna 1991 došlo ke značnému propadu ekonomiky, kvůli velké závislosti na odbytu především zemědělských produktů na velkém sovětském trhu.

V 90. letech začali vznikat významné hospodářské reformy, které přinesly významné změny ve vnímání některých ekonomických pojmů. Byly jimi například soukromé vlastnictví, hospodářská soutěž a podnikání. V současné době je podnikání uznáno všemi příslušnými subjekty, je rozvíjeno podle moldavského zákona a je jedním z významných a široce používaných pojmů.

V současnosti Moldavsko prochází reformou veřejné správy a decentralizací usiluje o ekonomické posílení a hospodářskou stabilitu. S touto reformou Moldavsku napomáhají příslušné mezinárodní organizace. Nicméně země se stále potýká s poměrně četnou řadou problémů. Velkým problémem je zmíněná šedá ekonomika, se kterou se slabé státní orgány a instituce nedovedou popasovat. Podstatným problémem je rovněž korupce, kde dochází k silnému sepětí mezi politickým a podnikatelským prostředím. Během transformace veřejné správy Moldavska byly identifikovány některé další nedostatky, jejichž náprava se jeví jako nezbytná zejména pro posílení právní jistoty subjektů právních vztahů. Těmito nedostatky je zejména nejednotnost terminologie, chybějící nebo nepřesně definované názvosloví.

# **1 CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ**

Cílem mé bakalářské práce je vytvoření programu, který umožní porovnávat podobnost, respektive vzdálenost, dvou slov v několika jazycích. Tento program bude v budoucnu využit při tvorbě slovníku s přesným názvoslovím a bude určený pro pracovníky veřejné správy v Moldavsku. Porovnávaná slova budou v českém, anglickém a rumunském jazyce, který byl koncem roku 2013 schválen nejvyšším soudem jako úřední jazyk Moldavska.

Dílčím cílem je analýza různých algoritmů, vhodných pro určení míry podobnosti, respektive určujících vzdálenost dvou slov a také analýza prostředí Moldavské republiky.

## 2 TEORETICKÁ VÝCHODISKA PRÁCE

V této části práce se budu zabývat teoretickými východisky, které přiblíží problematiku popisovaného tématu.

### 2.1 Veřejná správa

Veřejnou správou z pohledu společnosti zorganizované ve stát, se rozumí správa veřejných záležitostí. Ta je realizována jako projev výkonné moci ve státě, respektive moci veřejné, kterou disponuje stát a jím schválené subjekty zaměřující se na správu veřejných záležitostí. Veřejná moc se dělí na moc státní a zbývající moc veřejnou [1].

Co se státní moci týče, ta je zabezpečována státem prostřednictvím zvláštního aparátu. Zbývající veřejná moc, jenž je v příslušném rozsahu státem svěřena ke správě veřejných záležitostí subjektů nestátního charakteru, je od státní moci odvozena a nemůže s ní být v rozporu. Z toho důvodu má zbývající veřejná moc svůj základ ve státní moci. Mají společné znaky a projevuje se jako decentralizovaná státní moc, jejíž subjekty společně se státem spoluzabezpečují správu státu, jakožto správu veřejnou [1].

Z hlediska zájmů je třeba odlišit veřejnou správu od správy soukromé. „Veřejná správa, správa veřejných záležitostí, je správou ve veřejném zájmu a subjekty, které ji vykonávají, ji realizují jako právem uloženou povinnost, a to z titulu svého postavení jako veřejnoprávních subjektů [2].“

Správa soukromá je naopak, jak její název vypovídá, správou soukromých záležitostí. Je vykonávána v soukromém zájmu, tedy soukromými subjekty, které sledují vlastní cíl a řídí se vlastní vůlí. Na výkonu veřejné správy se však mohou podílet i soukromoprávní subjekty. Subjekty veřejné správy mají možnost zadat výkon veřejných služeb soukromoprávním subjektům. Jejich činnost však není považována za soukromou, ale je správou veřejnou. To je z důvodu, že je vykonávána ve veřejném zájmu a při jejím provádění se musí řídit příslušnými právními předpisy. Takový výkon veřejné správy je možný jen v situaci, kdy postavení veřejnoprávního subjektu je zastupitelné a kdy to nevylučuje povaha věci [3].

### 2.1.1 Činnosti veřejné správy

Na úrovni státní činnosti je možné veřejnou správu definovat jako státní činnost, kterou stát vykonává vedle zákonodárství a soudnictví [4].

Veřejná správa se od zákonodárství liší tím, že veřejnou správou se zákony provádějí, načež zákonodárstvím se zákony vytvářejí. Nicméně i samy orgány veřejné správy se podílí na přípravě zákonů, jsou tedy subjekty tvorby práva. Jejich funkce nemůže být v rozporu s funkcí zákonodárných orgánů. V případě tvorby práva se nejedná o zákonodárství, ale o tzv. odvozenou normotvorbu. Ta představuje vydávání obecně závazných pravidel chování ve formě správních aktů normativní povahy [4].

Činnost soudnictví se od veřejné správy liší tím, že představuje nalézání práva ve sféře veřejných i soukromých zájmů, zatímco veřejná správa realizuje pouze veřejné zájmy (v mezích práva) [4].

Tím bylo popsáno funkční rozlišení, nicméně vedle toho je významné i rozlišení organizační. Soudnictví je prováděno orgány, které jsou si vzájemně rovny, a které i přes hierarchické uspořádání jejich soustavy (vztah podřízenosti a nadřízenosti) jsou na sobě naprosto nezávislé. Oproti tomu soustava orgánů veřejné správy je, vedle prvků rovnosti orgánů v rámci jejich příslušné úrovně, přísně ovládána a provázána vztahy podřízenosti a nadřízenosti. Soustava soudů se tedy vyznačuje koordinací a soudce je při výkonu zákonů nezávislý, zatímco soustava státních orgánů veřejné správy se vyznačuje subordínací a správní orgán je vázán příkazy výše postavených právních orgánů [4].

V rovině činnosti dalších veřejnoprávních subjektů (odlišných od státu) je veřejná správa vymezena jako nestátní činnost výkonného charakteru, jejímž základním cílem je spoluzabezpečovat výkon státní správy. Dalšími veřejnoprávními subjekty jsou veřejnoprávní korporace. Ty svoji subjektivitu odvozují od státu a současně jsou jím aprobovány. Tyto korporace vykonávají veřejnou správu svým jménem a ve své působnosti. Správa jimi vykonávána je svojí povahou samosprávou. Veřejná správa vykonávaná veřejnoprávními korporacemi zpravidla vykazuje stejné rysy jako veřejná správa vykonávaná státem, resp. jeho orgány [4].

### 2.1.2 Funkce veřejné správy

Funkce veřejné správy jsou vyjádřením požadavků a potřeb jejího fungování. Tyto funkce jsou konkrétním projevem, slouží k realizaci funkcí státu v podmínkách veřejné správy a jsou upraveny správním právem. Je možné je rozdělit na funkce obecné, tj. funkce veřejné správy vlastní veřejné správě jako celku, a funkce dílčí, tj. funkce jednotlivých organizačních subsystémů či orgánů veřejné správy [5].

V rámci obecné funkce plní veřejná správa dvě základní funkce:

- funkci organizující
- funkci mocenské ochrany

Hlavní podstatou organizující funkce veřejné správy je komplexní postižení procesů, které spočívají v uspořádání subjektů a adresátů působení veřejné správy a vztahů mezi nimi, postavení a vztahů vůči jiným subjektům [5].

Podstatou funkce mocenské ochrany je poskytování ochrany ze strany orgánů veřejné správy s cílem zajištění jejího bezporuchového chodu a zjednáání nápravy na místech, kde došlo k porušení právních povinností. Tato funkce je však omezená prostředky, které má veřejná správa pro plnění svých úkolů k dispozici. V případech, kdy situace přesahuje možnosti samotné veřejné správy, zajišťují ochranu k tomu povolané orgány. Nicméně nejde pouze o ochranu následnou nápravou, ale především o preventivní pozitivní vliv tohoto působení [5].

Uvedené základní obecné funkce veřejné správy se v procesu reálného fungování veřejnoprávního působení vzájemně prolínají a doplňují [5].

Dílčí funkce veřejné správy jsou chápány a vyjadřovány jako relativně konkrétní činnosti vykonávané veřejnou správou. V tomto pojetí lze vymezit funkce programovací, ovlivňování, koordinační, kooperační, rozhodovací a funkce kontrolní. Význam těchto funkcí spočívá v tom, že jejich prostřednictvím se při zcela konkrétním působení fakticky realizují obecné funkce veřejné správy. Při každém působení veřejné správy tak lze vysledovat jak obecnou, tak i konkrétní dílčí funkci veřejné správy [5].

### 2.1.3 eGovernment

Pojem eGovernment vyjadřuje sled několika procesů, které poskytují výkon veřejné správy a uplatňují občanská práva a povinnosti jak fyzických, tak právnických osob prováděných elektronickou formou. Jedná se tedy o modernizaci veřejné správy za pomoci informačních a komunikačních technologií. Obecně lze říci, že je to tzv. elektronizace veřejné správy. Hlavní myšlenkou pro zavedení této změny je, aby správa věcí veřejných byla za využití moderních elektronických nástrojů k občanům přátelštější, dostupnější, efektivnější, rychlejší a levnější [12].

Úloha eGovernmentu je definována jako optimální komunikace mezi [13]:

- úřady a občany (G2C - Government-to-Citizen)
- úřady samotnými (G2G - Government-to-Government)
- úřady a podniky (G2B - Government-to-Business)
- úřady a zaměstnanci (G2E - Government-to-Employees)

Modernizace veřejné správy – moderní veřejná správa by měla mít tyto vlastnosti [13]:

- získání informací od občana pouze jednou – obíhají data nikoli občan
- nabídka jednoho kontaktního místa pro všechny služby veřejné správy – možnost vyřídit vše na jednom místě; občan si může vybrat způsob komunikace
- poskytování proaktivních služeb – veřejná správa se snaží vyřešit vše bez přítomnosti občana – kontaktuje ho až v nezbytně nutných případech

Významnou myšlenkou eGovernmentu je také zjednodušit občanovi komunikaci s orgány veřejné a státní správy. Což vede ke snížení nároků na čas, který je potřebný k vyřízení požadovaných dokumentů. Elektronizace by měla zesílit a zefektivnit výkonnost státní správy, proto je nutné zajistit a podpořit činnost správních úřadů při realizování úkolů státní správy i samosprávy. Z toho důvodu je zapotřebí především vytvořit pravidla komunikačního prostředí, která se shodují s charakterem a obsahem úloh v kompetencích státních orgánů. Vytvořená pravidla tak podpoří chtěnou výměnu

informací a budou nevyhnutelným předpokladem pro spolupráci jednotlivých informačních systémů ve veřejné správě [6].

V neposlední řadě je rovněž cílem eGovernmentu vymezit procesně-správní povahu působení správních úřadů a jejího odrazu v úlohách informačních systémů. Dále je nutné myslet na vzájemné vztahy, které plynou z právního rámce fungování a zabezpečení předávání informací na základě vymezených práv a povinností [6].

Aby byla zaručena správná funkčnost eGovernmentu, je důležitá elektronizace interních agend ve veřejné správě. Tento úkol je v současnosti stále objektem hledání vhodného a efektivního východiska nejen v České republice, ale také v zahraničí [6].

Mezi hlavní překážky rozvoje eGovernmentu byla a je vnímána obecně nedokonalost legislativy, která na jednu stranu zaváděla nové instituty a nástroje, na druhou stranu však ponechávala v platnosti zastaralá organizačně technická pravidla. Další překážkou je také neexistence bezvýznamového identifikátoru osoby (dosud je identifikátorem rodné číslo), dále pak nerovnoprávnost formy listinné s formou elektronickou a nedostatečné vedení elektronických spisů a elektronické spisové služby [6].

#### **2.1.3.1. Výhody a nevýhody eGovernmentu**

Pokud se budeme dívat na eGovernment z různých úhlů, bude zjevné, že má své výhody i nevýhody.

- Výhody [13]:
  - dostupnost informací, veřejná správa 24 hodin 7 dní v týdnu
  - úspora času (žádné čekání na úřadech), zaměstnanců a peněz (na nákladech na provoz)
  - efektivnost, rychlost, kontinuita
  - pohodlí občana – obíhají data, nikoliv občané
  - garance správnosti dat (není zde chybující lidský faktor)
  - ochrana úředníků (když úředník určitý referenční údaj přijme a pracuje s ním, je v zákoně zakotveno, že tak činí v dobré víře a nemůže být nikým popotahován za to, že tento údaj použil, i kdyby se později prokázalo, že tento údaj byl chybný)



- Nevýhody [13]:
  - možnost nabourání se do systémů a zneužití dat
  - vysoké počáteční investice do softwaru, školení, atd.
  - nutná počítačová gramotnost - problém pro důchodce
  - pomalý vývoj – což je nákladné a například krušné začátky vzbuzují nedůvěru občanů (např. registr vozidel)

## 2.2 Formální jazyky

V roce 1956 byly položeny základy novodobé historie disciplíny formálních jazyků americkým matematikem Noamem Chomskym. Ten se zabýval studiem přirozených jazyků a v souvislosti s tím vytvořil matematický model gramatiky jazyka [14].

Původní představou bylo formalizovat popis přirozeného jazyka tak, aby byl umožněn automatizovaný překlad z jednoho přirozeného jazyka do druhého nebo aby tento jazyk sloužil jako prostředek komunikace člověka s počítačem. Nicméně realizace téhle představy se ukázala jako velmi obtížná a ani výsledky v současnosti nejsou uspokojivé [14].

Formální jazyky lze obecně definovat jako množinu slov nad určitou abecedou, která je vytvořena určitou gramatikou. Jedná se tedy o jakousi formalizaci jazyka, který je běžně chápán jako prostředek komunikace. Jelikož se jedná o poměrně obecný pojem, zahrnuje komunikaci na různých úrovních, např. mezilidská komunikace, programování, atd. Je tedy důležité definovat potřebné pojmy [15].

### 2.2.1 Abeceda

Abeceda je označení pro libovolnou konečnou množinu, jejíž prvky nesou název znaky (případně také písmena nebo symboly) abecedy. Abecedou tedy můžeme rozumět například znaky latinské abecedy (a až z), číslice (0 až 9), grafické symboly (např. Braillovo písmo) nebo vizuální symboly (např. barvy semaforu). Často se označuje velkým řeckým písmenem  $\Sigma$  (sigma) [15].

### 2.2.2 Řetězec

Řetězcem nebo také slovem nad abecedou  $\Sigma$ , rozumíme konečnou posloupnost znaků této abecedy. Tento pojem nese označení  $\omega$  (omega). Konečný počet znaků této posloupnosti, respektive konečnou délku slova značíme  $|\omega|$ . Je možné se však setkat i s prázdnou posloupností slova (tzv. prázdné slovo), která nese označení  $\varepsilon$  (epsilon). U takového slova platí, že jeho délka je rovna nule [15].

**Prefix** – tímto termínem se označuje slovo nebo část slova, která je umístěna na začátek jiného slova [16].

**Suffix** – tento termín označuje písmeno nebo skupinu písmen, která jsou přidána na konec slova [16].

### 2.2.3 Gramatika

Gramatika představuje množinu syntaktických pravidel, podle kterých je možné ověřit, jestli nějaké slovo patří do jazyka popsaného danou gramatikou, či nikoliv. Dále lze pomocí gramatiky vygenerovat všechna slova, pokud je jejich množina konečná [17].

Formální gramatika se značí  $G$  a skládá se ze čtveřice  $(N, T, P, S)$ , kde [17]:

- $N$  představuje konečnou množinu neterminálních symbolů
- $T$  představuje konečnou množinu terminálních symbolů ( $T \cap N = \emptyset$ )
- $P$  představuje konečnou množinu odvozovacích pravidel  
 $(T \cup N)^* N (T \cup N)^* \rightarrow (T \cup N)^*$
- $S$  představuje počáteční symbol ( $S \in N$ )

## 2.3 Metody porovnávání slov

V této kapitole budou předmětem teoretická východiska možnosti porovnávání podobností slov. Nejdříve bude představena Levenshteinova vzdálenost, která bude následně využita v praktické části práce. Dále pak bude popsána ještě Damerau-Levenshteinova vzdálenost a Hammingova vzdálenost.

### 2.3.1 Levenshteinova vzdálenost

Levenshteinova vzdálenost (Levenshtein Distance) je metoda, která určuje podobnost mezi dvěma řetězci znaků. Tuhle metodu osvětluje v roce 1966 ruský matematik Vladimir Levenshtein ve svém textu „Binary codes capable of correcting deletions, insertions, and reversals“ [7].

Často je možné setkat se s pojmem „editační vzdálenost“ neboli „změnová vzdálenost“. Z toho je patrné, že metrika udává vzdálenost podobnosti dvou řetězců, kde vzdálenost je dána nejmenším počtem znakových operací, které jsou potřebné ke změně jednoho řetězce na druhý. Levenshteinova vzdálenost obsahuje operace [8]:

- Operace identita
  - $hodnota(i, j) = hodnota(i-1, j-1)$
  - Využívá se, pokud jsou na průsečíku textů (i, j) shodné znaky
  - Operace identita odpovídá diagonálnímu pohybu v tabulce
- Mazání znaku (deletion)
  - $hodnota(i, j) = hodnota(i, j-1) + 1$
  - Zajistí odstranění znaku
  - Odpovídá horizontálnímu pohybu v tabulce
- Vkládání znaku (insertion)
  - $hodnota(i, j) = hodnota(i-1, j) + 1$
  - Zajistí vložení znaku do řetězce
  - Odpovídá vertikálnímu pohybu v tabulce

- Substituce znaku (substitution)
  - $hodnota(i, j) = hodnota(i-1, j-1) + 1$
  - Slouží pro nahrazení daného písmene v řetězci písmenem jiným
  - Odpovídá pohybu po diagonále

Abychom využili minimální počet operací, je nutné dbát na to, aby byla použita právě ta operace, která zajistí, že aktuální *hodnota* (*i*, *j*) bude minimální [8].

Výsledná hodnota Levenshteinovi vzdálenosti se nachází v matici v pravém dolním indexu [8].

Pro názornou ukázkou jsem vybral dvě náhodná slova, na kterých provedu výpočet Levenshteinovi vzdálenosti. V příkladu budu poměřovat slova „migration“ a „migrace“.

i/j	-	M	I	G	R	A	T	I	O	N
-	0	1	2	3	4	5	6	7	8	9
M	1	0	1	2	3	4	5	6	7	8
I	2	1	0	1	2	3	4	5	6	7
G	3	2	1	0	1	2	3	4	5	6
R	4	3	2	1	0	1	2	3	4	5
A	5	4	3	2	1	0	1	2	3	4
C	6	5	4	3	2	1	1	2	3	4
E	7	6	5	4	3	2	2	2	3	4

Tab. 1 – Matice po výpočtu LD

#### Postup průchodu maticí:

1. Shoda znaku „M“ a „M“, hodnota operace 0
2. Shoda znaku „I“ a „I“, hodnota operace 0
3. Shoda znaku „G“ a „G“, hodnota operace 0
4. Shoda znaku „R“ a „R“, hodnota operace 0
5. Shoda znaku „A“ a „A“, hodnota operace 0

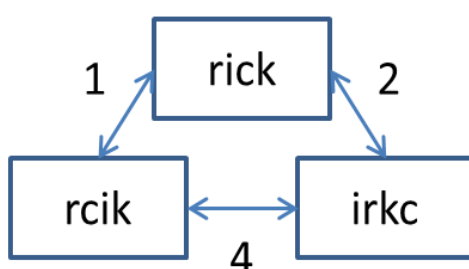
6. Vložení znaku „T“ do řetězce, hodnota operace 1
7. Vložení znaku „I“ do řetězce, hodnota operace 1
8. Substituce znaku „C“ za „O“, hodnota operace 1
9. Substituce znaku „E“ za „N“, hodnota operace 1

Výsledná Levenshteinova vzdálenost mezi řetězci je hodnota 4, což říká, že je zapotřebí čtyř změn, abychom dostali shodné řetězce.

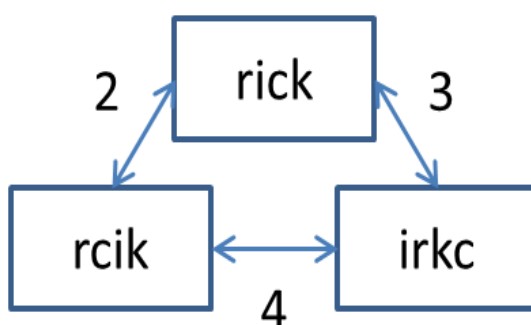
### 2.3.2 Damerau-Levenshteinova vzdálenost

Jak z názvu metody vyplývá, Damerau-Levenshteinova vzdálenost vychází ze samotné Levenshteinovi vzdálenosti. Mimo základní operace, popsané v předchozí kapitole, což byly operace mazání, vkládání a substituce, je tahle metoda doplněna o operaci *transpozice*. Samotné slovo *transpozice* lze chápat jako změnu vzájemné polohy [10].

Damerau-Levenshteinova vzdálenost tedy počítá s možností překlepů, respektive se záměnou dvou po sobě jdoucích znaků při psaní elektronického textu [10].



Obr. 1 – Damerau-Levenshteinova vzdálenost [9]



Obr. 2 – Levenshteinova vzdálenost [9]

### 2.3.3 Hammingova vzdálenost

Hammingova vzdálenost je pravděpodobně nejjednodušší metoda porovnávání dvou řetězců. Původní myšlenkou této metriky bylo, aby sloužila k detekci a korekci chyb během digitální komunikace, která probíhá za pomoci binárního kódu. Nicméně v praxi je možné ji využít i pro řetězce složené z textových znaků. Porovnávané řetězce však musí být stejně dlouhé nebo je nutné definovat speciální znak  $\lambda$ , kterým se doplní kratší slovo. Jednoduchost Hammingovi vzdálenosti také spočívá v tom, že zná pouze operaci substituce, tedy nahrazení znaku [11].

$s_1$	M	Ě	S	Í	C	
$s_2$	M	Ě	Š	E	C	
<i>substitute</i>	0	0	1	1	0	2

*Tab. 2 – Hammingova vzdálenost*

Z příkladu, ve kterém jsem zvolil dvě náhodná slova, vyplývá sama jednoduchost této metody. Hodnota Hammingovi vzdálenosti je v tomhle případě rovna číslu 2.

## 2.4 Visual Basic for Application (VBA)

VBA je objektově orientovaný programovací jazyk, který byl vytvořen společností Microsoft. Tato společnost jej zabudovala do svého univerzálního kancelářského balíčku Microsoft Office, kde se vyskytuje jako součást jednotlivých programů. Pro účely této práce se však budeme zabývat pouze využitím VBA v prostředí MS Excel [18].

MS Excel slouží pro vytváření tabulek s daty, které následně můžeme formátovat, tvořit z nich grafy, kontingenční tabulky, nebo můžeme vytvářet funkce pro práci s těmito daty. Taková práce však může být relativně časově náročná a mnohokrát se opakuje. Z důvodů pohodlnosti a urychlení činnosti nabízí MS Excel možnost vytváření maker [19].

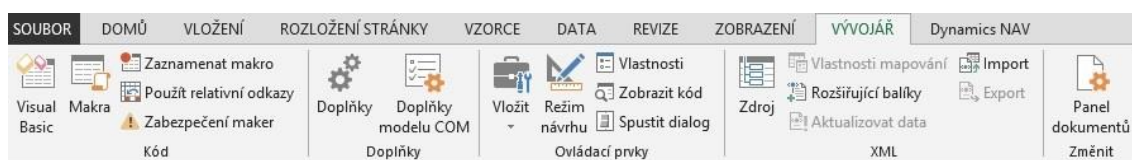
Makra napomáhají k automatizaci několika operací, které je jinak nutné několikrát opakovat. Tato makra je možné vytvářet dvojím způsobem [19]:

- uživatelsky – stačí pouze zapnout zvláštní režim, za pomoci kterého se zaznamenají všechny provedené akce pomocí VBA kódu.
- programátorsky – uživatel spustí editor, který je součástí programu, a v něm bude zapisovat potřebný kód.

Programátorský způsob vytváření maker však nabízí více možností pro zpracování dat. Je zde možné vytvářet vlastní funkce, zpracovávat data na listu jednodušším způsobem, spouštět automatické akce či vytvářet vlastní dialogová okna [19].

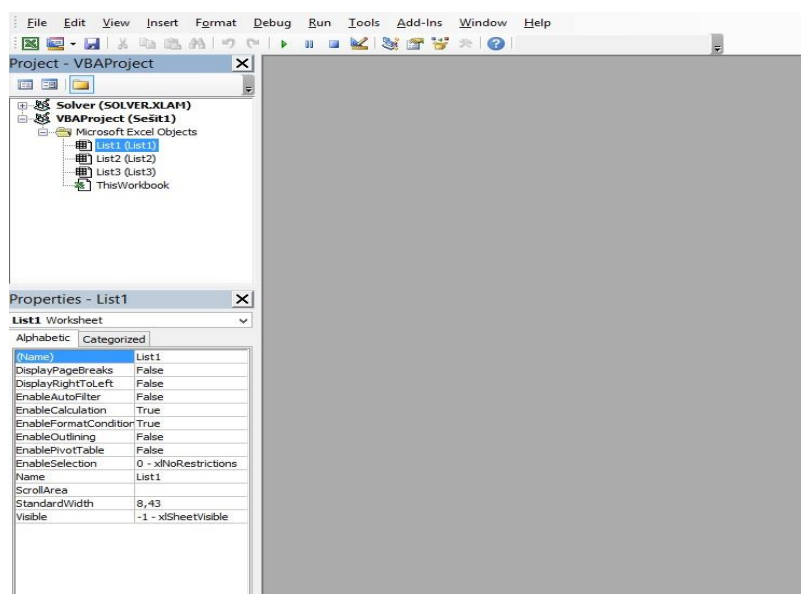
### 2.4.1 Vývojové prostředí Microsoft Visual Basic for Applications

Pro využití VBA je důležité přidat si do pásu karet kartu „Vývojář“, která nabízí různá nastavení pro vytváření maker, vstup do vývojového prostředí VBA nebo možnost vkládání aktivních prvků, kterým se následně přiřazuje funkcionality v již zmíněném vývojovém prostředí.



Obr. 3 – MS Excel, karta „Vývojář“

V samotném vývojovém prostředí je pak možné vytvářet vlastní funkce, formuláře, kód, který bude udávat funkcionalitu aktivních prvků umístěných na kterémkoliv z listů daného sešitu nebo upravovat vlastnosti různých objektů.



Obr. 4 – Vývojové prostředí VBA

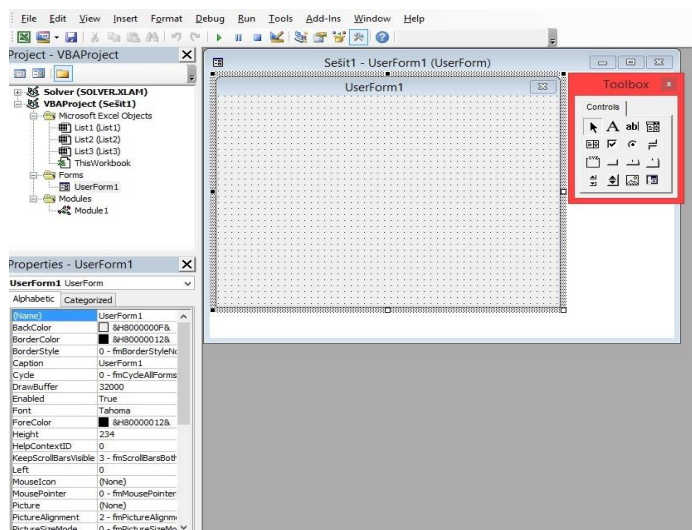
Okno „Properties“ neboli vlastnosti nabízí veškeré vlastnosti objektu, který je zrovna označený. Je zde tedy možné upravovat vlastnosti jednotlivých listů nebo celého sešitu, formulářů a aktivních prvků. Větší význam však má upravovat vlastnosti aktivních prvků, kde můžeme měnit díky tomuto oknu například styl písma, název, text na aktivním prvku, podbarvení, umístění, rozměry, atd.

Přes nabídku „Insert“, která se nachází v horní liště vývojového prostředí, je možné vložit formulář nebo modul pro vytváření funkcí. Při vložení formuláře se zobrazí návrhové okno, ve kterém si uživatel upravuje vzhled samotného formuláře, přičemž do něj zároveň vkládá aktivní prvky pro jeho ovládání a jim přiřazuje potřebný kód.

Pokud vytváříme formulář, máme dvě možnosti jeho spuštění. Buď se formulář spustí při spuštění listu, na kterém je umístěný, nebo je nutné vložit příkazové tlačítko,



kterému vytvořený formulář přiřadíme. V obou případech je možné spustit formulář modálně nebo nemodálně, což znamená, že u modálního zobrazení je možné ovládat pouze formulář, dokud není zavřen, kdežto u nemodálního zobrazení je možné i přes otevřený formulář editovat buňky a tak upravovat data.



*Obr. 5 – Vytvoření nového formuláře*

Při vytvoření formuláře se zároveň otevře i okno s názvem „Toolbox“. Jedná se o okno, které uživateli nabízí možnost vkládání jednotlivých aktivních prvků, jako jsou například:

- Label – slouží pro nadpis
- TextBox – okno pro vkládání textu
- CommandButton – aktivační tlačítko
- CheckBox – zaškrtnávací okno
- OptionButton – přepínač



*Obr. 6 – Nabídka aktivních prvků pro formulář*

### 3 ANALÝZA SOUČASNÉHO STAVU

V této kapitole se budu zabývat analýzou současného stavu Moldavska. Zejména tedy veřejnou správou země a chybějící odbornou terminologií pro její efektivní fungování. Dále se zde budu zabývat dostupností aplikací, službách pro výpočet Levenshteinovi vzdálenosti.

#### 3.1 Analýza současného stavu Moldavska

Moldavská republika je východoevropským státem, který leží mezi Ukrajinou a Rumunskem. Počet obyvatel je necelé 4 miliony občanů a hlavním městem je Kišiněv. Úředním jazykem je moldavština, ale užívá se zde také ruština a rumunština.

Co se administrativního dělení Moldavské republiky týče, je složena z:

- 32 okresů
- 3 statutární města (Kišiněv, Bălți, Tighina)
- 1 autonomní oblast (Gagauzsko)
- 1 autonomní oblast se zvláštním statusem (Podněstří)

Každý okres volí radu, která dohlíží a koordinuje činnosti místních rad v oblasti poskytování veřejných služeb. Tyto rady jsou voleny podle tamního práva na dobu 4 let.

##### 3.1.1 Místní veřejná správa

Systém místní veřejné správy se skládá ze dvou úrovní:

1. úroveň – veřejné subjekty s obecnými nebo zvláštními pravomocemi vytvořenými a fungujícími na území obce nebo města s cílem prosazování zájmů místní komunity a řešením svých problémů.
2. úroveň - veřejné subjekty s obecnými nebo zvláštními pravomocemi vytvořenými a fungujícími na území okresů a hlavního města Moldavské republiky Kišiněva a zvláštním právním postavením autonomní územní jednotky s cílem prosazování zájmů a řešení problémů obyvatel dané územní správní jednotky.

V souladu s právními předpisy, činnost první a druhé úrovně místních veřejných orgánů podléhá správní kontrole. Zákon o místní veřejné správě poskytuje seznam položek, které jsou předmětem povinného dozoru vlády a přístrojů. Správní kontrola zahrnuje kontrolu právních předpisů a přiměřenost politických orgánů veřejné správy.

Státní kancelář je zodpovědná za organizaci administrativní kontroly zákonnosti činnosti místních orgánů veřejné správy. To vykonávána vlastními orgány nebo prostřednictvím podřízených územních úřadů.

Ústava Moldavska, jakož i zákon o místní veřejné správě tvrdí, že vzájemné vztahy orgánů veřejné správy jsou založeny na principech autonomie, zákonnosti, transparentnosti a spolupráce při řešení společných problémů.

Podle zákona o místní veřejné správě okresní rady mohou rozhodnout o spolupráci s ostatními orgány místní veřejné správy, včetně přeshraniční spolupráce, na uskutečnění činností a provedení nezbytných veřejných služeb, jakož i jako spolupráci s národními a zahraničními hospodářskými subjekty a nevládními organizacemi vzhledem k realizaci činností nebo prací se společnými zájmy.

### **3.1.2 Problémy ve veřejné správě**

Pochybný stav služeb v odpovědnosti druhé úrovně místní veřejné správy a zmatek mezi touto úrovní veřejné správy a decentralizovanými službami. Správní legislativa a praxe Moldavské republiky není jasně stanovena, což je zásadní rozdíl mezi decentralizovanými službami.

Různé zprávy, analyzující celkový stav ve veřejném sektoru Moldavské republiky, podtrhují přehnaný růst, týkající se zejména veřejných výdajů v hrubém národním produktu (HNP). Ve srovnání s úrovní rozvoje navrhuje omezení jeho rozšíření. Tento faktor má zvláštní vliv na proces decentralizace z důvodu extrémního omezení fiskálního prostoru, nezbytného pro lepší financování místní veřejné správy.

### 3.1.3 Kompetence

Zákon o územní veřejné správě definuje vlastní a přenesené pravomoci okresů. Mezi povinnosti okresů patří:

- Sociální, hospodářský, územní a městský rozvoj
- Výstavba okresních silnic, výstavba veřejných zařízení v okrese
- Poskytování vzdělávacích institucí
- Poskytování sociální pomoci
- Poskytování sportovních a jiných aktivit
- Ochrana životního prostředí
- Správa majetku
- Služby požární ochrany, atd.

K dispozici je také široká škála povinností přenesených na okresy ze strany státu, včetně sociální ochrany, zdravotní péče, veřejné bezpečnosti apod., v těchto oblastech jsou místní orgány v souladu se zákonem předmětem správního dozoru.

Podle práva mají úřady místní veřejné správy (subjekty 1. úrovně) autonomii, nicméně ta je poněkud omezená. Z části je to proto, že orgány veřejné správy 2. úrovně je ruší v každodenní činnosti. Důvodem je, že jejich finanční limit nedosahuje potřeb. Tento nedostatek je doprovázen nadměrně roztržitou územně-správní organizací a rovněž z důvodu nedostatečné správní a institucionální kapacity.

Tyto podmínky představují nízkou odpovědnost místní veřejné správy 1. stupně. Vymezení pravomoci mezi první a druhou úrovní místní veřejné správy je nejasné, protichůdné a v některých případech dokonce chybí. Některé činnosti uvedené jako in-house kompetence nejsou jasně definované, a tak jsou často bezprostředně přisuzovány vyšší úrovni. Tato situace umožňuje dvojí, tím pádem nejednoznačný, výklad odpovědností či pravomocí různých úrovní administrativně-územních jednotek. Zde nejsou jasné a funkční technická kritéria, která by byla přijata a použita výslovně pro definování, vymezení, zajištění kompetencí a financování. Nedostatek jasného vymezení kompetencí má tendenci vyvíjet tlak a tím destabilizovat celý systém administrativních vztahů.

### **3.1.4 Financování**

Zákon o místní veřejné správě uvádí, že orgány místní veřejné správy podléhají finanční autonomii. Podle zákona o místních veřejných financích rozpočet okresu obsahuje:

- Příjmy a výdaje potřebné k plnění povinností přiřazené k okresu podle právních předpisů a dalších povinností přenesených k vládě
- Místní rozpočty, které se skládají z rozpočtů vesnic a měst v okrese

V Moldavské republice, stejně jako v mnoha jiných tranzitivních zemích, decentralizace některých veřejných služeb a převod některých odpovědností či pravomocí nebyla splněna v souladu s převodem zdrojů, které jsou nezbytné pro jejich plnění. To vede ke vzniku hluboké nerovnováhy ve financování. V důsledku toho je obtížné a někdy i nemožné vykonávat činnost menších administrativně-územních jednotek nebo těch, kteří mají nízký rozpočet. Proto je nutné zajistit mechanismus, který zaručí administrativně-územním jednotkám nezbytné střednědobé a dlouhodobé finanční zdroje pro plnění odpovědností či pravomocí přenesených ze strany státu.

### **3.1.5 Problémy s jazykem**

Problematika odborné terminologie úzce souvisí s vývojem jazykového prostředí Moldavska. V tomto ohledu je stále ještě patrný vliv blízké historie. Moskevská vláda protlačovala od roku 1941, aby byl moldavský jazyk odlišný od jazyka rumunského. Ve stejné době bylo latinské písmo nahrazeno ruskou azbukou. Média, školství a úřady začaly nuceně používat jazyk, vytvořený z moldavských nářečí z oblasti Transnistrie. Až po smrti J. V. Stalina v roce 1953 se začala uplatňovat taktika přibližování se k rumunštině z druhé strany Prutu, avšak zásadní změny se neuskutečnily. Velký posun zpět k rumunskému jazyku nastal až v letech 1988-1989. V tomto období vyšlo přibližně 800 různých článků, jejichž autoři vyzývali k návratu latinské abecedy. Otázka jazyka je v Moldavsku stále žhavým tématem. Na jedné straně stojí proti-rumunská část obyvatel proti nemalé části populace, která se mimo jiné ve sčítání lidu v roce 2004 označila jako „Rumuni“. Takové označení zvolilo až 30% populace Moldavska.

Jednou z hlavních priorit moldavské vlády je posilování transparentnosti a efektivnosti veřejné správy a veřejných služeb v Moldavsku. Během probíhajících změn

byly odhaleny závažné nedostatky, týkající nejednotnosti terminologie. To je patrné zejména v okamžiku, kdy je snaha sjednotit moldavské zákony a předpisy s přímo použitelnými předpisy Evropské unie.

Sjednocení terminologie by výrazně přispělo ke zvýšení efektivnosti výkonu veřejné správy, ke zvýšení slučitelnosti národní úpravy s právem Evropské unie, k doplnění nebo zpřesnění některých institutů a zejména ke zvýšení právní jistoty v zemi.

K tomu Moldavsku napomáhá nejen Evropská unie, ale i jiné mezinárodní organizace. Jistý podíl na spolupráci a podpoře Moldavska má i společnost DCB Actuaries and Consultants s.r.o. Jedná se o českou společnost sídlící v Brně.

### **3.2 DCB Actuaries and Consultants s.r.o.**

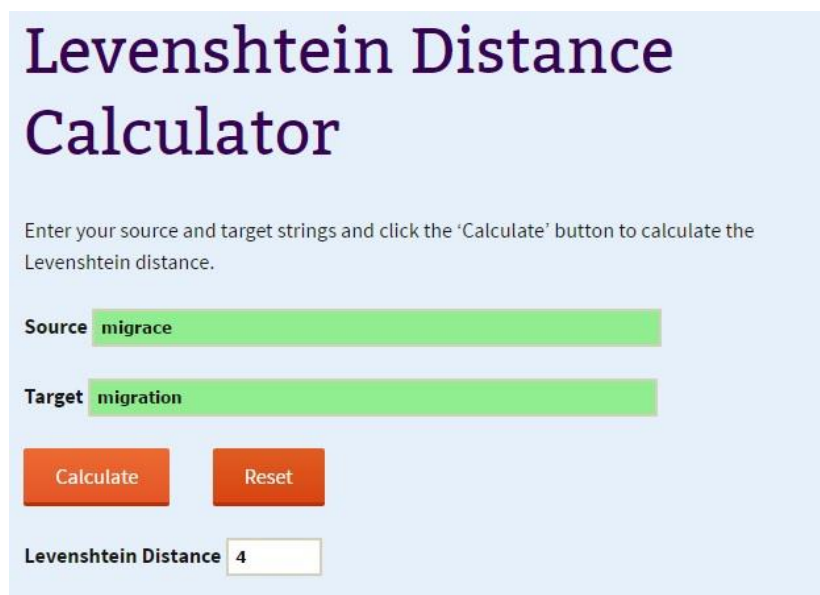
Tato společnost se zabývá dodávkou komplexních řešení pro klienty v oblastech zdravotnictví, pojišťovnictví, penzijních fondů a státní správy. S ohledem na oblast státní správy se společnost momentálně podílí i na projektu tvorby systému pro správu elektronických dokumentů i návrh řešení jejich archivace ve veřejné správě v Moldavsku. Pro jeho efektivní fungování je však důležité sjednotit terminologii, která, jak bylo popsáno výše, není v Moldavsku úplně jednotná. Tamní úřady používají různé termíny, ač myslí shodnou věc, v některých případech termín pouze popisují.

Z toho důvodu je nutné v první řadě vytvořit slovník pojmů, který bude v rámci státní správy používán. V první fázi budou identifikovány termíny, které jsou používány jednotně. Následně budou další odborné termíny zkoumány co do duplicity. V případě, že vhodný termín v jazyce neexistuje, je nutné ho definovat. Během tohoto procesu je v určité fázi nutné zjistit podobnost dvou slov z různých jazyků, přičemž pro porovnávání termínů z oblasti státní správy je v současné době anglický jazyk. Výsledkem porovnání bude vzdálenost dvou termínů. Nakonec podle těchto výsledků bude možné vybrat termíny, které jsou shodné, případně co nejvíce blízké a vytvořit slovník s přesně definovanými termíny.

### 3.3 Dostupnost aplikace

Možností pořízení programu pro měření Levenshteinovi vzdálenosti je hned několik. Je však důležité si ujasnit podmínky, které jsou důležité pro pořízení takového programu.

První možností je najít si volně dostupný online program, kterých je na internetu dostupných víc než dost. Pak už jen stačí zadat porovnávaná slova a potvrzením si nechat vypsat jejich vzdálenost. Je to jednoduché a pro obvyčejného uživatele bezstarostné. Na druhou stranu online program nemusí vždy dávat správný výsledek. Dále také z hlediska bezpečnosti uchování dat je vhodné se poohlédnout po jiném než online řešení.



*Obr. 7 – online kalkulátor Levenshteinovi vzdálenosti [20]*

Druhou možností pak může být vyhledání a kontaktování specializované firmy, která dokáže na zakázku naprogramovat to, co uživatel požaduje. Nicméně to může být poměrně finančně náročné.

Třetí a poslední možností je vytvořit si něco sám. Jedinou potřebou je mít alespoň základní znalosti některého programovacího jazyka a za pomoci dostupných materiálů, ať už učebnic nebo volně dostupných informací na internetu, si vytvořit program dle vlastních potřeb. Z hlediska bezpečnosti dat se jedná o ideální řešení, kdy má uživatel všechna data pod svou kontrolou.

## 4 VLASTNÍ NÁVRH ŘEŠENÍ

V této části se budu zabývat praktickou aplikací Levenshteinovi vzdálenosti na Česko-anglický a Anglicko-rumunský slovník za pomoci využití programu Microsoft Excel 2013.

### 4.1 Naprogramování funkce

Levenshteinova vzdálenost a prostředí Microsoft Visual Basic for Applications bylo popsáno již výše. Nyní se tedy budeme zabývat naprogramováním této funkce.

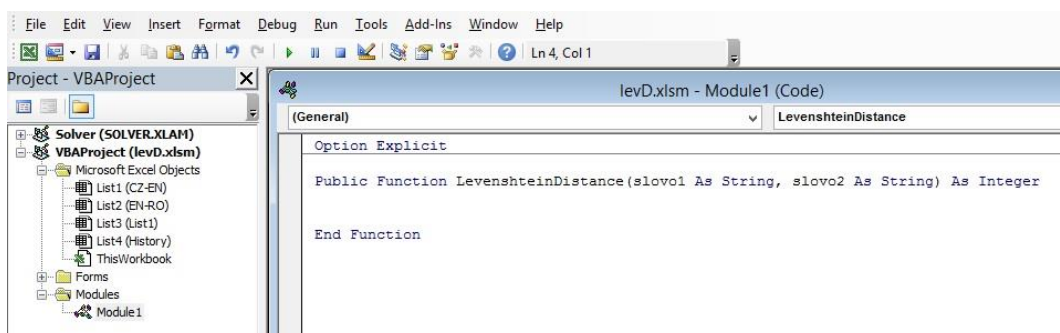
Do vývojového prostředí se uživatel dostane přes kartu „Vývojář“, ta sice není primárně přednastavena a uživateli se tak nezobrazuje, nicméně je možné si ji mezi přednastavené karty přidat. Kliknutím na tlačítko „Soubor“ se v levém sloupci zobrazí nabídka. V té uživatel zvolí volbu „Možnosti“, která povede k zobrazení okna „Možnosti aplikace Excel“ a v sekci „Přizpůsobit pás karet“ stačí nalézt požadovanou kartu a přidat ji mezi tzv. hlavní karty. Poté stačí okno potvrdit, čímž se vrátíme zpět na list sešitu s tím, že máme mezi kartami jednu novou.

Ve vývojovém prostředí VBA si musíme přes nabídku „Insert“ vložit nový modul, do kterého budeme vytvářet samotný kód naší funkce. Modul obsahuje takzvané veřejné proměnné, což znamená, že to co se v něm naprogramuje, je možné využít kdekoliv v celém projektu. Proto bude možné vytvořenou funkci Levenshteinovi vzdálenosti použít ve kterémkoliv listu.

Prvním příkazem, před psaním samotné funkce, je příkaz „Option Explicit“, ten zajišťuje nutnost deklarovat veškeré proměnné, s kterými budeme pracovat.

Funkce obecně funguje tak, že obsahuje nějaké vstupní proměnné, na základě kterých proběhne nějaký kód a nakonec zobrazí výslednou hodnotu této funkce. V našem případě jsou vstupními proměnnými dvě slova, která budeme porovnávat a zjišťovat jejich vzdálenost. Zmíněná vzdálenost pak bude výstupní hodnota funkce a bude číselného charakteru.



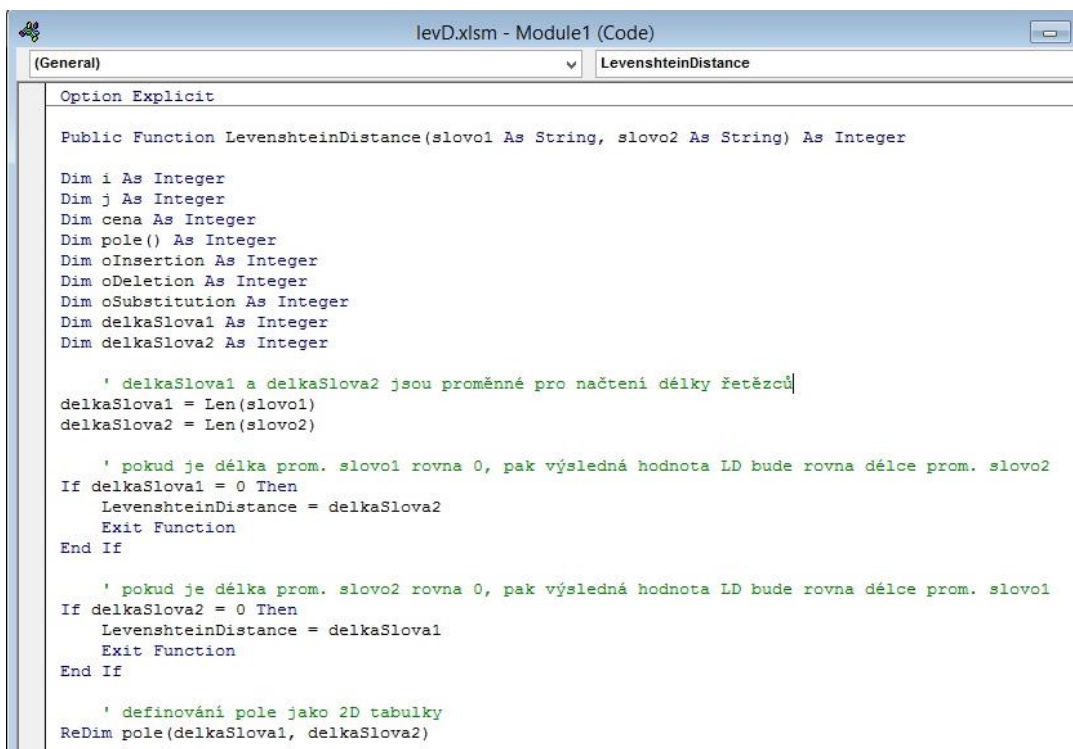


*Obr. 8 – Základ pro programování funkce*

Dalším krokem bude definování proměnných, které budeme používat v těle funkce. V mém případě jsem si navrhl devět proměnných, z nichž jedna bude sloužit jako dvourozměrné pole, které je možné představit si jako tabulku, kterou jsme využili v ukázkovém příkladu na Levenshteinovu vzdálenost v teoretické části práce. V případě tohoto pole se jedná o pole dynamické, které má takovou vlastnost, že se deklaruje až v běhu programu, z toho důvodu mu při deklaraci proměnných určíme pouze to, že se jedná o pole a jeho datový typ.

Nyní začne hlavní fáze programování funkce. Zde je v první řadě důležité zjistit délku slov vstupních proměnných do funkce. Tuhle hodnotu zjišťujeme hned z několika důvodů. Prvním z nich je, že v případě, pokud je délka jednoho ze slov rovna nule, pak výsledkem této funkce bude hodnota délky slova druhého a celá funkce tak může skončit a nepokračovat v průchodu dalšího kódu. Druhým důvodem pak je, že zde budeme deklarovat rozměry pole, které do této doby bylo nedefinované. Jelikož funkce bude porovnávat dvě slova, bude se jednat o dvourozměrné pole, kde počet řádků bude odpovídat jednomu slovu a počet sloupců druhému slovu. Pro deklaraci tohoto pole v těle programu je důležitý příkaz „ReDim“, pomocí kterého program ví, co se bude dít.

V dalším kroku opět využijeme proměnné, do kterých jsme uložili hodnoty délek jednotlivých slov. Za pomoci cyklu načteme do již vytvořeného pole základní hodnoty, od kterých se budou odrážet hodnoty Levenshteinovi vzdálenosti. Jsou to hodnoty od nuly do maximální hodnoty délky slova. Tyto hodnoty se budou zapisovat jak na každý řádek, tak na každý sloupec. Proto zde budou využity dva cykly.



```
levD.xlsm - Module1 (Code)
(General) LevenshteinDistance
Option Explicit

Public Function LevenshteinDistance(slovo1 As String, slovo2 As String) As Integer

    Dim i As Integer
    Dim j As Integer
    Dim cena As Integer
    Dim pole() As Integer
    Dim oInsertion As Integer
    Dim oDeletion As Integer
    Dim oSubstitution As Integer
    Dim delkaSlova1 As Integer
    Dim delkaSlova2 As Integer

    ' delkaSlova1 a delkaSlova2 jsou proměnné pro načtení délky řetězců
    delkaSlova1 = Len(slovo1)
    delkaSlova2 = Len(slovo2)

    ' pokud je délka prom. slovo1 rovna 0, pak výsledná hodnota LD bude rovna délce prom. slovo2
    If delkaSlova1 = 0 Then
        LevenshteinDistance = delkaSlova2
        Exit Function
    End If

    ' pokud je délka prom. slovo2 rovna 0, pak výsledná hodnota LD bude rovna délce prom. slovo1
    If delkaSlova2 = 0 Then
        LevenshteinDistance = delkaSlova1
        Exit Function
    End If

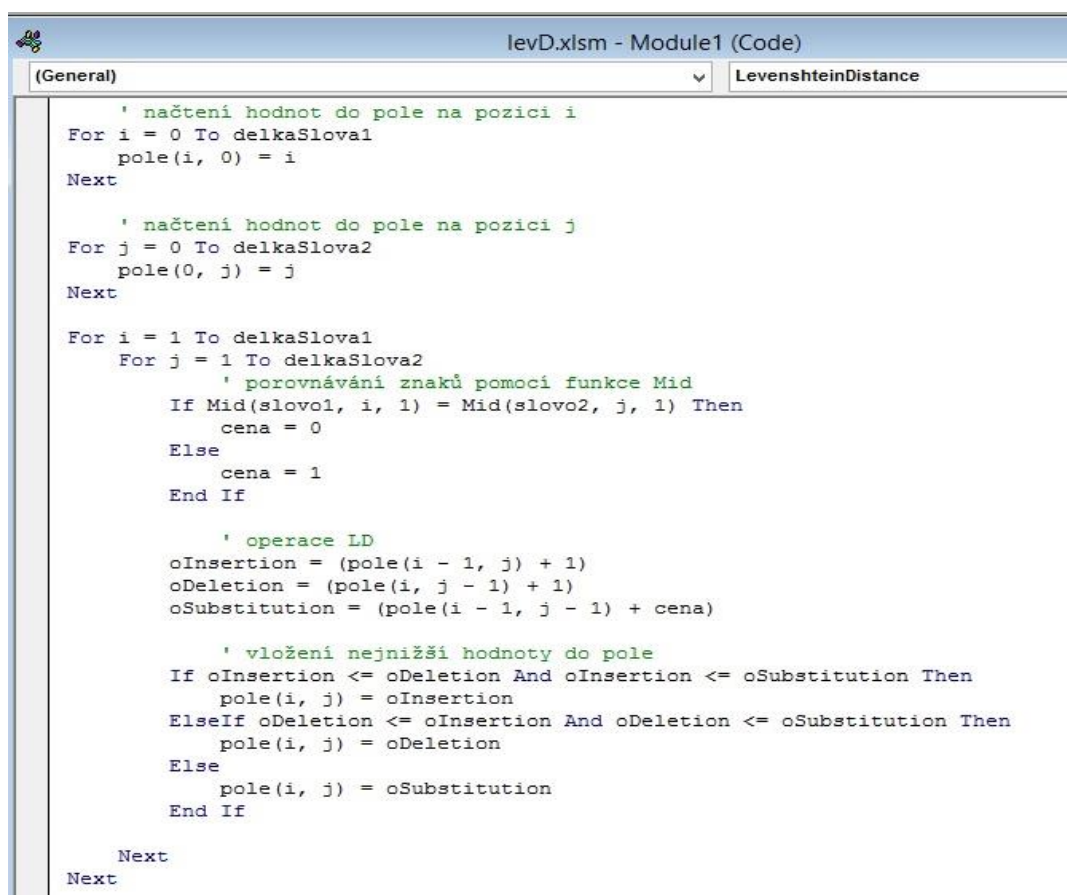
    ' definování pole jako 2D tabulky
    ReDim pole(delkaSlova1, delkaSlova2)
```

Obr. 9 – první část funkce

Po zapsání základních hodnot do pole můžeme začít praktikovat podstatu Levenshteinovi vzdálenosti. Budeme tedy porovnávat každý znak jednoho slova se znakem na stejné pozici ve druhém slově. K tomu využijeme opět cykly, nyní však budou začínat od jedničky a končit v maximální hodnotě délky slova. Druhý cyklus bude vnořený do cyklu prvního, což zajistí, aby se porovnávaly znaky na shodné pozici. Vnořený cyklus pak bude obsahovat podmínku, ve které využijeme funkci Excelu „Mid“. Tato funkce slouží k tomu, aby vracela zadaný počet znaků ze zadaného řetězce od zadané pozice. Pomocí toho jsme tedy schopni porovnávat znaky z obou řetězců. Podmínka je postavena tak, že pokud jsou znaky, zjištěné funkcí „Mid“, shodné, bude se proměnná odpovídající ceně operace Levenshteinovi vzdálenosti rovnat nule, pokud se však bude jednat o znaky rozdílné, bude se stejná proměnná rovnat hodnotě jedna. Tím bude tato podmínka ukončena.

Následně nastavíme výchozí hodnoty jednotlivých operací Levenshteinovi vzdálenosti. Jedná se o operace „Insertion“ (vkládání), „Deletion“ (mazání) a „Substitution“ (záměna), které jsou blíže specifikované v teoretické části práce v podkapitole „Levenshteinova vzdálenost“.

Jelikož je nutné zajistit, aby bylo využito minimálního počtu operací, musí být aktuální hodnota Levenshteinovi vzdálenosti rovněž minimální. Proto po zajištění výchozích hodnot jednotlivých operací bude následovat složená podmínka, ve které budeme zjišťovat, která z operací má nejnižší hodnotu. Znění podmínky je následovné: pokud hodnota operace „Insertion“ je rovna nebo menší jak hodnoty operace „Deletion“ a zároveň operace „Substitution“, bude do dříve vytvořeného pole zapsána právě jeho hodnota; pokud tomu tak nebude, podmínka zjistí, zdali operace „Deletion“ je menší nebo rovna jak hodnoty operací „Insertion“ a zároveň „Substitution“, v tom případě zapíše do pole tuhle hodnotu; pokud ani tohle nebude vyhodnoceno jako pravda, pak se do pole zapíše hodnota operace „Substitution“. Tím se ukončí podmínka a proběhne další kolo obou cyklů, dokud nejsou cykly naplněny.



```

levD.xlsm - Module1 (Code)
(General) LevenshteinDistance

' načtení hodnot do pole na pozici i
For i = 0 To delkaSlova1
    pole(i, 0) = i
Next

' načtení hodnot do pole na pozici j
For j = 0 To delkaSlova2
    pole(0, j) = j
Next

For i = 1 To delkaSlova1
    For j = 1 To delkaSlova2
        ' porovnávání znaků pomocí funkce Mid
        If Mid(slovo1, i, 1) = Mid(slovo2, j, 1) Then
            cena = 0
        Else
            cena = 1
        End If

        ' operace LD
        oInsertion = (pole(i - 1, j) + 1)
        oDeletion = (pole(i, j - 1) + 1)
        oSubstitution = (pole(i - 1, j - 1) + cena)

        ' vložení nejnižší hodnoty do pole
        If oInsertion <= oDeletion And oInsertion <= oSubstitution Then
            pole(i, j) = oInsertion
        ElseIf oDeletion <= oInsertion And oDeletion <= oSubstitution Then
            pole(i, j) = oDeletion
        Else
            pole(i, j) = oSubstitution
        End If
    Next
Next

```

Obr. 10 – druhá část funkce

Po naplnění cyklů už pak proběhne pouze volání funkce, která vypíše výsledek, v tomto případě vzdálenost dvou zadaných slov, což znamená konec procesu funkce.

```

' volání a výsledek vytvořené funkce
LevenshteinDistance = pole(delkaSlova1, delkaSlova2)

End Function

```

Obr. 11 – třetí část funkce

Tím je programování funkce Levenshteinovi vzdálenosti u konce a nyní je možné ji začít aplikovat.

## 4.2 Aplikování funkce na slovníky

Od společnosti DCB Actuaries and Consultants s.r.o. mi byly poskytnuty slovníky, na kterých bude použita mnou naprogramovaná funkce Levenshteinovi vzdálenosti. Jednalo se o slovníky Česko-anglický a Anglicko-rumunský. V této části jsem využil vypsání výsledku funkce za pomoci vzorce v dané buňce listu. Do prázdné buňky, do které bude vypsána výsledná vzdálenost funkce, se vloží znak „rovná se“ (=), za něj název požadované funkce, v mém případě „*LevenshteinDistance*“, a do závorky údaj, ze kterých buněk budou použita slova. Funkce očekává dvě buňky, které budou odděleny středníkem. Poté stačí potvrdit výběr stisknutím klávesy „enter“ a vypíše se vzdálenost dvou vstupních slov.

SUMA :    =LevenshteinDistance(B9;C9)				
	A	B	C	D
1		Slovička česky	Slovička anglicky	LD
2	3.01	přístup	access	7
3	3.02	odpovědnost	accountability	12
4	3.03	sledování činností	action tracking	15
5	3.04	archiv	archival authority	12
6	3.05	třídění	classification	14
7	3.06	klasifikační systém	classification system	8
8	3.07	konverze	conversion	5
9	3.08	zničení	destruction	=LevenshteinDistance(B9;C9)

Obr. 12 – výpočet Levenshteinovi vzdálenosti přes vzorec

Pro ulehčení práce Excel nabízí uživateli možnost roztáhnutí vzorce do několika řádků pod sebou. Stačí tak mít vypsané slovíčka a vzorec napsat pouze do prvního řádku. Následně při označení buňky s výsledkem vzdálenosti stačí najet kurzorem na pravý dolní roh buňky a podržením kliknutí na něj roztáhnout výsledek až do

posledního požadovaného řádku. Excel pak vypíše všechny vzdálenosti bez nutnosti opět zadávat vzorec.

Tuhle variantu rozšíření vzorce jsem aplikoval na obou slovnících, přičemž každý slovník jsem vložil na jiný list sešitu z důvodu přehlednosti. Na jednom listu jsem tak měl Česko-anglický slovník a Anglicko-rumunský slovník jsem měl na listu druhém.



Obr. 13 – listy sešitu

## 4.3 Uživatelské rozhraní

V další fázi jsem se zabýval tvorbou uživatelského rozhraní, respektive něčeho, co usnadní uživateli využívání tohoto programu.

### 4.3.1 Přidání nového záznamu do slovníku

Jako první jsem zvolil možnost přidávat nové záznamy do slovníku. Na každý list jsem vložil aktivní prvek „*CommandButton*“ neboli příkazové tlačítko, kterému jsem přiřadil uživatelský formulář. Nabídka aktivních prvků se nachází na kartě „*Vývojář*“ v sekci „*Vložit*“, ale kód pro takové tlačítko je nutné vytvářet ve vývojovém prostředí.

Formulář obsahuje dvě textové pole a tři příkazová tlačítka. V textových polích jsou očekávány textové řetězce, které se následně budou porovnávat. Funkčnost příkazových tlačítek je pak následující:

- první tlačítko slouží pro výpočet Levenshteinovi vzdálenosti, kde se po stisknutí tlačítka nejprve zobrazí okno se zprávou, která obsahuje výsledek funkce. Po potvrzení okna se záznam zapíše na první prázdný řádek ve slovníku.
- druhé tlačítko slouží k tomu, aby uživateli šetřilo čas při mazání textových řetězců v textových polích, takže při jeho stisknutí se textové pole vyprázdní.
- třetí tlačítko pak slouží k uzavření formuláře a navrácení zpět ke slovníku.

U formuláře jsem zvolil nemodální zobrazení, takže i přes jeho otevření je možné editovat buňky.





### 4.3.2 Ostatní doplňky u slovníků

Aby práce se slovníky byla pro uživatele pohodlná a nenáročná, umístil jsem k nim pár dalších doplňků. Listy se slovníky jsem obohatil o tlačítka „*Vymazání dat z označené oblasti*“ a „*Uložit do PDF*“.

Pomocí tlačítka „*Vymazání dat z označené oblasti*“ může uživatel mazat data z předem označené oblasti. Stačí pouze stisknutím levého tlačítka myši označit oblast, ze které se mají data odstranit a následně kliknout na tlačítko. Kromě toho, že tlačítko zvládne odstranit údaje v jednotlivých buňkách, zvládne odstranit i jakékoliv formátování těchto buněk, takže vrátí oblast do původního stavu.

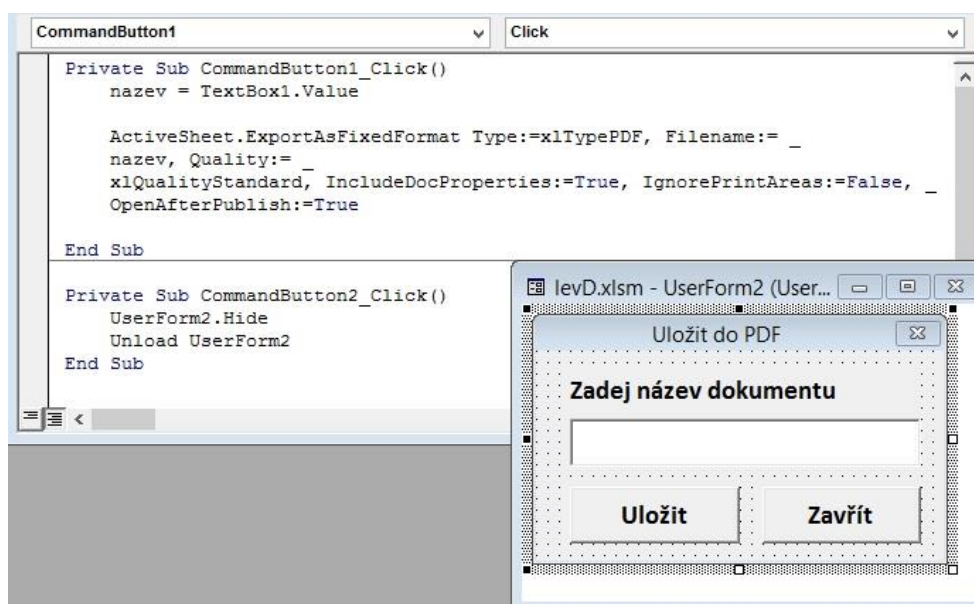
Pro vytvoření tohoto tlačítka stačí pouze umístit příkazové tlačítko na list souboru a přiřadit mu jednořádkový kód.



Obr. 16 – kód pro tlačítko „*Vymazání dat z označené oblasti*“

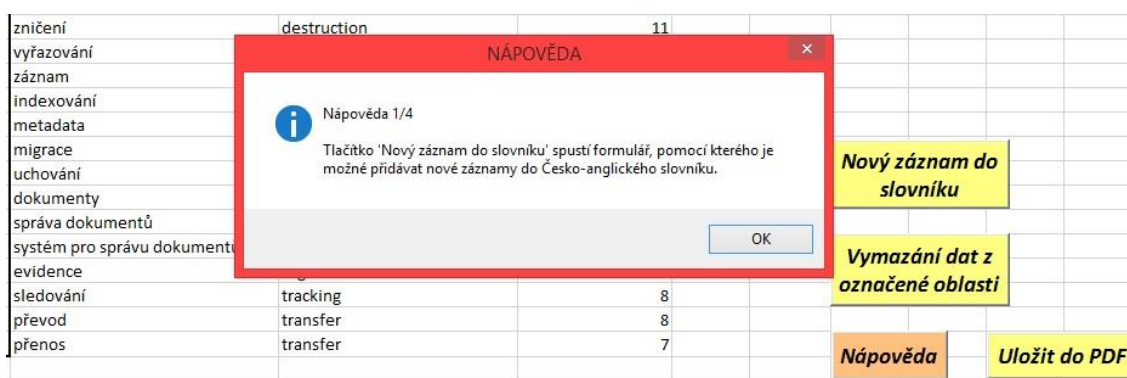
Dalším doplňkem je tlačítko „*Uložit do PDF*“. PDF z anglického „*Portable Dokument Format*“ v překladu znamená „*Přenosný formát dokumentu*“. Jedná se o software od společnosti Adobe, který je v dnešní době velmi rozšířený nejen kvůli své funkcionalitě, ale také proto, že je nabízen zdarma. Z toho důvodu uživatel tuhle možnost uložení všeho, co má na listu se slovníkem, do formátu PDF, jistě ocení.

Aby však uživatel mohl zadat název, pod kterým chce dokument uložit, po stisknutí tlačítka se zobrazí jednoduchý formulář, který obsahuje textové pole a dvě příkazové tlačítka. Do textového pole se запиše textový řetězec a pomocí prvního tlačítka „*Uložit*“ se soubor uloží do formátu PDF a zároveň se otevře v příslušném softwaru. PDF dokument se uloží na disk „*C*“ do složky „*Dokumenty*“. Druhé příkazové tlačítko formuláře pak slouží k uzavření tohoto formuláře.



Obr. 17 – Formulář a jeho kód pro zadání názvu a uložení listu do formátu PDF

Posledním doplňkem u slovníku je tlačítko, pod kterým se ukrývá nápověda. Ta může posloužit v případě, že by uživatel nevěděl, jak tyto slovníky ovládat. Při stisknutí tlačítka s nápovědou vyskočí okno, v němž se zobrazí první část nápovědy. Při stisknutí tlačítka „OK“ v tomto okně se uživatel dostane na další části.



Obr. 18 – Nápověda a přidaná příkazová tlačítka

Nápověda není vytvořena přes formulář, ale přes tzv. MsgBox, respektive okno se zprávou. Do tohoto okna je možné umístit různé tlačítka za pomoci různých příkazů, například: pouze OK (příkaz: „VbOKOnly“), tlačítka OK a Storno (příkaz: „VbOKCancel“) a další. Rovněž je možné vkládat ikony pro kritickou zprávu, varovnou zprávu, informační zprávu nebo varovný dotaz. V mém případě jsem zvolil tlačítko pouze OK a ikonu informační zprávy, která je ukryta pod příkazem „VbInformation“.



```

CommandButton6 Click
Private Sub CommandButton6_Click()
    i = MsgBox("Nápověda 1/4" & vbCrLf & "" & vbCrLf & "TEXT NÁPOVĚDY!", vbOKOnly + vbInformation, "NÁPOVĚDA")
    i = MsgBox("Nápověda 2/4" & vbCrLf & "" & vbCrLf & "TEXT NÁPOVĚDY!", vbOKOnly + vbInformation, "NÁPOVĚDA")
    i = MsgBox("Nápověda 3/4" & vbCrLf & "" & vbCrLf & "TEXT NÁPOVĚDY!", vbOKOnly + vbInformation, "NÁPOVĚDA")
    i = MsgBox("Nápověda 4/4" & vbCrLf & "" & vbCrLf & "TEXT NÁPOVĚDY!", vbOKOnly + vbInformation, "NÁPOVĚDA")

```

Obr. 19 – kód pro nápovědu

Příkaz „VbCrLf“, který je možné vidět v kódu, slouží k tomu, že umožní psát další část textu na nový řádek.

Jak možnost přidání nového záznamu do slovníku, tak všechny ostatní doplňky popsané v této kapitole jsou umístěny na obou listech. Každý list se slovníkem má tedy shodné ovládání.

U obou slovníků jsem nakonec využil i možnost vložení filtru, jenž je součástí aplikace Excel. Filtr je možné nalézt na kartě „Domů“ v sekci „Úpravy“ pod záložkou „Seřadit a filtrovat“. Nejdřív je nutné označit oblast, na kterou se bude daný filtr vztahovat. V našem případě stačí označit první řádek listu v prvních čtyřech sloupcích, tedy oblast hlavičky slovníku. Poté se rozklikne záložka „Seřadit a filtrovat“ a zde se zvolí možnost „Filtr“. V každém sloupci v označené oblasti se zobrazí šipka, kterou je možné rozkliknout a zvolit, podle čeho se mají záznamy seřadit.

	A	B	C	D	E	F	G	H	I	J
1		Slovička česky	Slovička anglicky	LD						
2	3.01	přístup	access	7						
3	3.02	odpovědnost	accountability	12						
4	3.03	sledování činností	action tracking	15						
5	3.04	archiv	archival authority	12						
6	3.05	třídění	classification	14						
7	3.06	klasifikační systém	classification system	8						
8	3.07	konverze	conversion	5						
9	3.08	zničení	destruction	11						
10	3.09	vyřazování	disposition	11						
11	3.10	záznam	document	8						
12	3.11	indexování	indexing	4						
13	3.12	metadata	metadata	0						
14	3.13	migrace	migration	4						
15	3.14	uchování	preservation	11						
16	3.15	dokumenty	records	9						
17	3.16	správa dokumentů	records management	14						
18	3.17	systém pro správu dokumentů	records system	23						
19	3.18	evidence	registration	10						
20	3.19	sledování	tracking	8						
21	3.20	převod	transfer	8						
22	3.21	přenos	transfer	7						
23										
24										

Přepínač mezi listy

**Úvodní list**

**EN - RO slovník**

**Historie porovnávaných slov**

**Nový záznam do slovníku**

**Vymazání dat z označené oblasti**

**Nápověda** **Uložit do PDF**

Obr. 20 – konečný vzhled slovníku

#### 4.3.3 Přidání nových listů do sešitu

V dalším kroku jsem rozšířil sešit o dva nové list, které nesou označení „Úvodní list“ a „Historie“. Jak z obou názvů vyplývá, „Úvodní list“ plní funkci jakési úvodní

stránky celého programu a „*Historie*“ slouží k ukládání záznamů, které chce uživatel uchovat, ale nechce je zapisovat do některého ze slovníků.

Protože nyní sešit obsahuje již 4 listy, vytvořil jsem pomocí příkazových tlačítek přepínač. Na každém listu tak přibyly nové prvky, které slouží pro pohybování se mezi jednotlivými listy. Tyto tlačítka mají za úkol, aby uživateli nabídly pohodlnější pohyb v rámci celého dokumentu.



Obr. 21 – přepínač mezi listy

Pod těmito tlačítky je ukryt jednoduchý jednořádkový kód. Každé tlačítko pod sebou skrývá příkaz „*Worksheets (“název\_listu”).Select*“, kde v závorce je název listu, na který má dané tlačítko odkazovat.

#### 4.3.4 Úvodní list

Na úvodním listě je poskládaný objekt z aktivních prvků, který slouží k porovnávání slov s možností zapisování záznamů buď do některého slovníku, nebo do historie, nebo se záznam neuchová, tedy se nikam nezapiše a pouze vrátí výslednou hodnotu vzdálenosti. V tomto objektu jsou využity následující prvky:

- dvě textové pole
- tři zaškrťovací pole
- dvě tlačítka
- dva prvky sloužící pro popis
- celý objekt je ohraničen prvkem „skupinový rámeček“

V textových polích se očekávají textové řetězce, které se budou porovnávat. V zaškrťovacích polích může uživatel označit ty pole, do kterých bude chtít záznam z porovnávání uložit. Je možné označit libovolný počet polí, tedy nula až všechny. Následně stiskne tlačítko pro Levenshteinovu vzdálenost, které zapíše výslednou hodnotu do buňky „G16“, případně do zvolených dalších listů. Stejně jako u formuláře na listech se slovníky i zde je tlačítko pro vyprázdnění textových polí. Pro jednodušší přesouvání tohoto objektu jsem označil všechny tyto prvky a využil možnosti seskupení těchto prvků. K této možnosti je možné se dostat tak, že uživatel označí prvky, které bude chtít seskupit a kliknutím na pravé tlačítko myši vyjede nabídka, kde zvolí možnost „seskupit“.

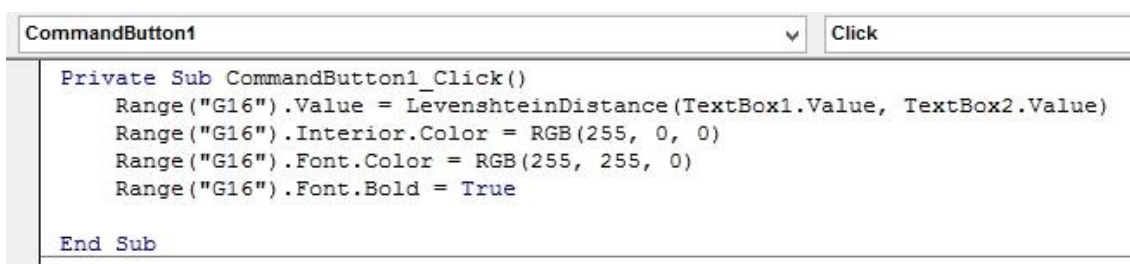
Obr. 22 – Objekt pro porovnávání slov na úvodním listě

Tento objekt může do jisté míry připomínat formulář, který byl vytvořený na listech se slovníky, nicméně v tomto případě se nejedná o formulář, ale pouze objekt poskládaný z aktivních prvků.

U tlačítka „Vyprázdnit“ je funkčnost podobná, jako obdobné tlačítko ve formulářích na listech se slovníky, i zde má za úkol pouze vyčistit textové pole, aby se uživatel nemusel zdržovat s mazáním textových řetězců, navíc ale vynuluje buňku, do které se zapisuje výsledná hodnota.

Nicméně u tlačítka „Levenshteinova vzdálenost“ je funkčnost o trochu složitější. I když tlačítko slouží rovněž pro výpočet vzdálenosti dvou slov, pomocí dříve vytvořené funkce, je zde navíc provázané se zaškrťovacími poli a výsledek funkce není vypsán

pomocí zobrazení okna se zprávou, ale zapíše se do mnou vybrané buňky. Tuto buňku jsem zvolil až poté, co jsem měl definitivně rozmístěné veškeré aktivní prvky. Jedná se o buňku „G16“, které jsem pomocí kódu nastavil nejen zobrazení výsledku vzdálenosti, ale i formátování, takže i když uživatel buňku například podbarví jinou barvou, při výpočtu Levenshteinovi vzdálenosti se buňka vybarví tak, jak jsem v kódu určil. Co se týče textového řetězce před buňkou s výsledkem, ten je zapsán pouze normálním způsobem, tedy zapsáním do buňky. Jen v případě tohoto textu jsem sloučil určitý počet buněk do jedné.



Obr. 23 – kód pro vypsání výsledku a formátování buňky G16

Abych se ale vrátil k tomu provázání tlačítka se zaškrťovacími poli. Uživatel má možnost uchovat záznam z výpočtu vzdálenosti a podle jeho důležitosti ho nechat uložit buď do některého slovníku, nebo do historie. Případně může tyto zaškrťovací pole kombinovat. Pokud využije uložení záznamu do nějakého slovníku, najde se v daném slovníku první prázdný řádek a do něj se záznam zapíše tak, jak je slovník postaven, tzn. dvě slova a hodnota výsledné vzdálenosti. Pokud však využije možnosti uložení do historie, zde se zapíše mimo slov a výsledku také datum a čas vytvoření záznamu. Údaje o datumu a času jsou použity ze systémového nastavení počítače uživatele. V listu s historií jsou rovněž pomocí kódu formátovány buňky, do kterých se bude zapisovat.

```

If CheckBox1 = True Then
For radek = 1 To 1000
    If Worksheets("CZ-EN").Cells(radek, 2) = "" And Cells(radek, 3) = "" And Cells(radek, 4) = "" And Cells(radek, 5)
        Worksheets("CZ-EN").Cells(radek, 2) = TextBox1.Value
        Worksheets("CZ-EN").Cells(radek, 3) = TextBox2.Value
        Worksheets("CZ-EN").Cells(radek, 4) = LevenshteinDistance(TextBox1.Value, TextBox2.Value)
        Exit For
    End If
Next radek
End If

```

Obr. 24 – kód pro provázání příkazového tlačítka se zaškrťovacím polem

Kód z obrázku 24 navazuje na kód z obrázku 23. Zde je patrné, že pokud je zaškrtnuté příslušné pole, respektive je mu určena hodnota „True“ tedy pravda, tak se

pomocí cyklu vybere řádek v listu. Záznamy lze přidávat do řádku 1000, nicméně je patrné, že jednoduchým zásahem do kódu lze tuto hodnotu změnit. Tento cyklus pak obsahuje podmínku, která zajistí, aby se záznam zapsal na úroveň jednoho stejného řádku, to pro případ, že by v předchozích řádcích někde chyběla hodnota. Dále je patrné, že se záznam bude zapisovat na list Česko-anglického slovníku.

Kód pro zapsání záznamu do Anglicko-rumunského slovníku pak vypadá shodně, jen je přiřazen dalšímu zaškrťovacímu poli.

U zápisu na list Historie je pak kód rozšířen právě o zmíněné formátování jednotlivých buněk, do kterých se záznam bude zapisovat. Navíc jsem pro tento list rozšířil počet řádků, do kterých je možné záznamy zapisovat, na hodnotu 3000, ale opět jednoduchým zásahem do kódu lze tuto hodnotu změnit.

```

If CheckBox3 = True Then
For radek = 1 To 3000
    If Worksheets("History").Cells(radek, 1) = "" And Cells(radek, 2) = "" And Cells(radek, 3) = "" And Cells(radek, 4) = "" Then
        Worksheets("History").Cells(radek, 2) = TextBox1.Value
        Worksheets("History").Cells(radek, 2).HorizontalAlignment = xlCenter
        Worksheets("History").Cells(radek, 3) = TextBox2.Value
        Worksheets("History").Cells(radek, 3).HorizontalAlignment = xlCenter
        Worksheets("History").Cells(radek, 4) = LevenshteinDistance(TextBox1.Value, TextBox2.Value)
        Worksheets("History").Cells(radek, 4).Interior.Color = RGB(0, 255, 0)
        Worksheets("History").Cells(radek, 4).HorizontalAlignment = xlCenter
        Worksheets("History").Cells(radek, 1).Value = Format$(Now, "dd/mm/yyyy hh:nn:ss")
        Worksheets("History").Cells(radek, 1).HorizontalAlignment = xlCenter
    End If
Exit For
End If
Next radek
End If

```

*Obr. 25 - kód pro provázání příkazového tlačítka se zaškrťovacím polem na list Historie*

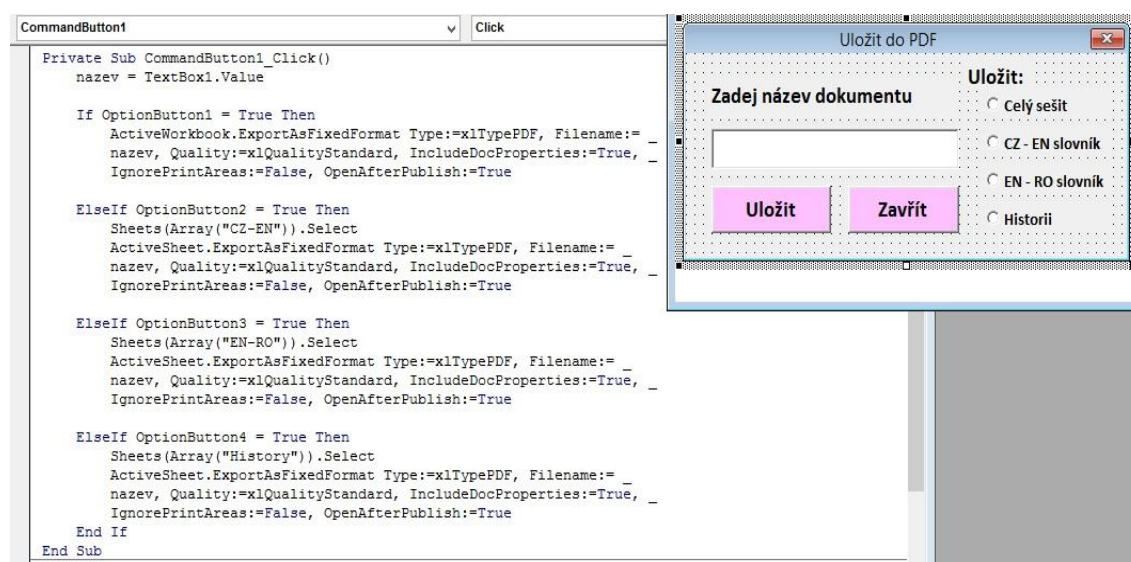
#### 4.3.5 Doplnky do Úvodního listu

V tuto chvíli máme na Úvodním listě pouze objekt pro porovnávání textových řetězců a tlačítkový přepínač pro pohyb mezi jednotlivými listy sešitu.

Na tento list jsem následně přidal pár doplňků, které uživateli mohou usnadnit práci s tímto programem. Jedná se o doplňky vytvořené pomocí příkazových tlačítek. Tato tlačítka pak uživateli nabízí možnost:

- uložit do PDF buď celý sešit, nebo jednotlivé listy
- uložit sešit ve výchozím formátu (formát .xlsm)
- uložit a zavřít sešit, ale nadále zůstat v aplikaci Excel
- ukončit celou aplikaci Excel
- nápověda

Tlačítko „*Uložit do PDF*“ již známe ze slovníků, nicméně zde se jedná o tlačítko s rozšířenou funkcí. I zde tohle tlačítko slouží k volání formuláře, který pak obsahuje jedno textové pole, dvě příkazové tlačítka a čtyři tlačítka „*OptionButton*“, tedy takzvaný přepínač. V textovém poli se opět očekává textový řetězec, který bude sloužit jako název nově vzniklého dokumentu formátu PDF. Dále pak uživatel za pomoci přepínače zvolí, co chce uložit, má na výběr buď celý sešit, nebo pouze jeden vybraný list sešitu kromě Úvodního listu. Pokud uživatel nevybere žádnou možnost z přepínače a stiskne tlačítko „*Uložit*“, nic se neprovede. V opačném případě, kdy zvolí, co chce uložit do formátu PDF a stiskne tlačítko pro uložení, proběhne vytvoření a uložení dokumentu, opět na disk „C“ do složky „*Dokumenty*“, a vytvořený dokument se uživateli otevře.



Obr. 26 – návrhové zobrazení formuláře a kód pro tlačítko „*Uložit*“

Druhým doplňkem Úvodního listu je tlačítko pro uložení sešitu ve výchozím formátu. Tento sešit je uložen s příponou „*XLSM*“ tedy sešit s podporou maker. Po stisknutí tlačítka „*Uložit sešit*“ se uloží veškeré provedené změny v tomto sešitě do stejného souboru. Tohle tlačítko pod sebou ukrývá jednořádkový kód „*ThisWorkbook.Save*“.

Dalším přidáním doplňkem pro jednodušší ovládání programu je tlačítko „*Uložit a zavřít sešit*“. Na rozdíl od tlačítka „*Uložit sešit*“, které slouží pouze k uložení provedených změn v sešitě, zde je možné sešit nejen uložit, ale zároveň ho zavřít. Tím se neukončí celá aplikace Excel, pouze se zavře sešit a aplikace jako taková zůstane



aktivní. Tlačítko pod sebou skrývá opět pouze jednořádkový kód „*ActiveWorkbook.Close True*“, kde příkaz „*True*“ udává, že se sešit má před uzavřením uložit.

V další fázi jsem pak přidal ještě tlačítko, které umožní uživateli ukončit celou aplikaci Excel. Pokud byly před stisknutím tohoto tlačítka provedeny v sešitu nějaké změny, sama aplikace se uživatele dotáže, zdali chce před ukončením uložit provedené změny. Stejně jako v předchozích dvou případech i tohle tlačítko si vystačí s jednořádkovým kódem „*Application.Quit*“.

Posledním doplňkem na tomto listu je příkazové tlačítko, které nese název „Nápověda“. Jak z názvu tlačítka vyplývá, stejně jako tomu bylo u listu se slovníky i zde se při stisknutí tohoto tlačítka objeví okno se zprávou, ve které je nápověda pro uživatele, jak program ovládat a k čemu vůbec slouží.

Nakonec, aby Úvodní list byl vizuálně přívětivější, skryl jsem mřížku listu, respektive Excelem přednastavené ohraničení jednotlivých buněk. Tahle možnost se ukrývá v kartě „*Rozložení stránky*“ v sekci „*Mřížka*“.

Obr. 27 – Úvodní list sešitu

#### 4.3.6 List Historie

Z obrázku 22 je patrné, že se porovnávají slova „*dokumenty*“ a „*records*“ přičemž záznam tohoto porovnání se má uložit do historie. Historie může být užitečná v případě, že uživatel nechce vložit záznam do slovníku, ale někdy v budoucnu, aby záznam

nemusel opakovat, si ho může vyhledat právě na tomto místě. Při ukládání do historie se ukládají textové řetězce, výsledná vzdálenost a datum a čas vytvoření záznamu.

	A	B	C	D
1	<b>Historie porovnávání slov</b>			
2	<b>Datum a čas</b>	<b>První slovo</b>	<b>Druhé slovo</b>	<b>LD</b>
3	03. 05. 2016 13:27:03	migration	migrace	4
4	03. 05. 2016 18:36:35	přístup	access	7
5	04. 05. 2016 18:22:34	dokumenty	records	9
6				
7				

Obr. 28 – ukázka záznamů z historie

Tento list sešitu je rovněž obohacen o doplňkové tlačítka. Jsou zde obsaženy tlačítka pro vymazání dat z označené oblasti a uložení listu do formátu PDF. Obě tlačítka i s kódy byly popsány již dříve.

	A	B	C	D	E	F	G	H
1	<b>Historie porovnávání slov</b>							
2	<b>Datum a čas</b>	<b>První slovo</b>	<b>Druhé slovo</b>	<b>LD</b>				
3	03. 05. 2016 13:27:03	migration	migrace	4				
4	03. 05. 2016 18:36:35	přístup	access	7				
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								

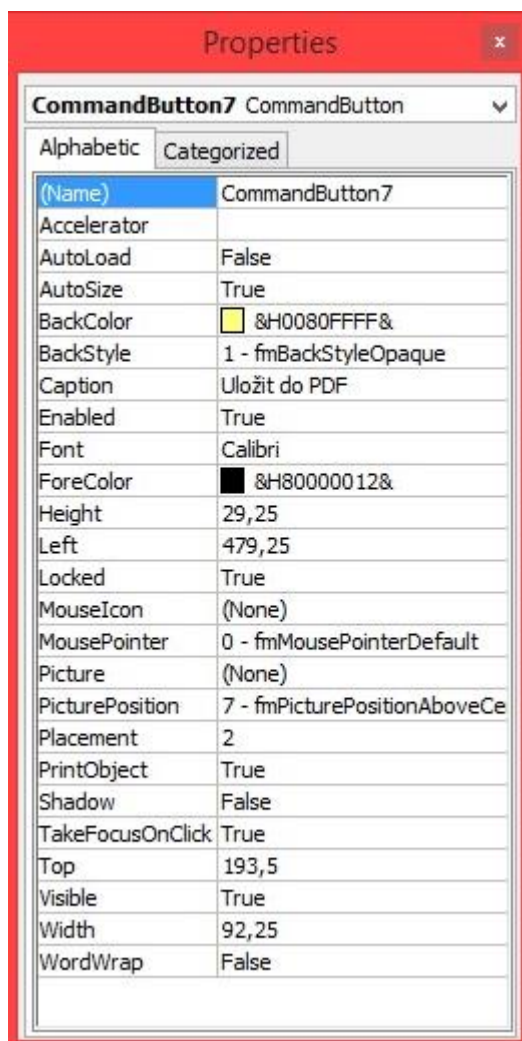
Obr. 29 – list Historie

#### 4.3.7 Vlastnosti aktivních prvků

Z celého programu je pak patrné, že aktivní prvky vložené na jakýkoliv list nebo i ve formulářích jsou upravené. Jedná se o úpravu vlastností těchto prvků. K vlastnostem se dostaneme tak, že aktivujeme na kartě „Vývojář“ možnost „Režim návrhu“ a kliknutím pravým tlačítkem myši na jakýkoliv aktivní prvek se zobrazí nabídka, ve které zvolíme možnost „Vlastnosti“, případně ve vývojovém prostředí VBA je to možnost „Properties“. V případě formuláře, tedy aktivních prvků ve vývojovém



prostředí VBA se zobrazí v levém panelu okno, kde lze vlastnosti upravovat. Pokud však upravujeme vlastnosti jakéhokoli prvku na listě sešitu, tak se zobrazí okno s vlastnostmi, které lze volně přesouvat kamkoliv na obrazovce.



Obr. 30 – okno pro upravování vlastností aktivních prvků

## ZÁVĚR

V této práci jsem analyzoval současnou situaci v Moldavsku, jenž je východoevropským státem. Jedná se o stát, který byl v minulosti součástí Sovětského svazu, jenž na něm zanechal značné stopy. I z toho důvodu je Moldavsko považováno za jednu z nejchudších zemí v Evropě.

Nicméně ze strany mnoha států i z jiných kontinentů než jen z Evropy a stejně tak moldavské vlády a různých mezinárodních institucí, je vidět snaha o zlepšení živobytí moldavských občanů. Probíhají různé reformy ve spravedlnosti, vzdělávání, ekonomice, energetice a v mnoha dalších sektorech.

Jak vyplynulo z analýzy, závažným problémem země je chybějící nebo nepřesná odborná terminologie. Díky tomu mají nejen moldavské úřady, ale i podnikatelské subjekty problém se mezi sebou vůbec dorozumívat a komunikovat spolu. Jakýkoliv rozvoj je tedy poměrně neefektivní a ekonomika, ač roste, tak roste pomalu.

Hlavním cílem mé práce bylo vytvořit program pro výpočet vzdálenosti dvou slov, jenž se dá v budoucnu využít pro vytvoření slovníku s jasně definovanými termíny. V práci jsem se zaměřil na vzdálenost odpovídajících termínů v českém, anglickém a rumunském jazyce. Výsledné vzdálenosti nabývaly různých hodnot, přičemž více shodných nebo alespoň velmi podobných slov se vyskytovalo v Anglicko-rumunském slovníku.

## SEZNAM POUŽITÉ LITERATURY

- [1] PRŮCHA, P., POMAHAČ, R. *Lexikon, správní právo*. 1. vyd. Ostrava: Sagit, 2002. 686 s. ISBN 80-7208-314-7.
- [2] KADEČKA, S., PRŮCHA, P. *Správní právo, obecná část*. 1. vyd. Brno: Masarykova univerzita v Brně, 2004. 74 s. ISBN 80-210-3600-1.
- [3] PRŮCHA, P. *Základy správního práva a veřejné správy, obecná část*. 2. přepr. a dopl. vyd. Brno: MU v Brně, 1994. 260 s. ISBN 80-210-0886-5.
- [4] PRŮCHA, P. *Správní právo, obecná část*. 6. dopl. a akt. vyd. Brno: DOPLNĚK, 2004. 360 s. ISBN 80-7239-157-7.
- [5] PRŮCHA, P., SCHELLE, K. *Základy místní správy*. 1. vyd. Brno: IURIDICA BRUNESIA, 1995. 192 s. ISBN 80-85964-00-7.
- [6] ŠTĚDRŮ, B. *Úvod do eGovernmentu v České republice: právní a technický průvodce*. 1.vyd. Praha: Úřad vlády České republiky, 2007. 172 s. ISBN 9788087041253.
- [7] LEVENSHTAIN, V. I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 1966, roč. 10, č. 8, s. 707–710.
- [8] Levenshteinova vzdálenost. *Algoritmy.net* [online]. 2015 [cit. 2016-03-10]. Dostupné z: <https://www.algoritmy.net/article/1699/Levenshteinova-vzdalenost>
- [9] Damerau-Levenshtein distance. *Richardminerich.com* [online]. [cit. 2016-03-10]. Dostupné z: <http://richardminerich.com/tag/damerau-levenshtein-distance/>
- [10] Damerau-Levenshtein distance. *Guyrutenberg.com* [online]. [cit. 2016-03-10]. Dostupné z: <https://www.guyrutenberg.com/2008/12/15/damerau-levenshtein-distance-in-python/>
- [11] Triviální algoritmus pro vyhledávání vzorku textu. *Ms.mff.cuni.cz* [online]. [cit. 2016-05-12]. Dostupné z: [http://www.ms.mff.cuni.cz/~kopecky/vyuka/dis/09/dis09\\_v5.html](http://www.ms.mff.cuni.cz/~kopecky/vyuka/dis/09/dis09_v5.html)
- [12] Co je to eGovernment? *Ministerstvo vnitra České republiky* [online]. [cit. 2016-03-10]. Dostupné z: <http://www.mvcr.cz/clanek/co-je-egovernment.aspx>

- [13] PŘENOSIL, Jan. *Veřejná správa – eGovernment*. [přednáška]. Brno: PEF Mendelu, 5. března 2015.
- [14] ČEŠKA, M., T. VOJNAR a A. SMRČKA. *Teoretická informatika* [online]. VUT FIT, Brno, 2014 [cit. 2016-05-12]. Dostupné z: <http://www.fit.vutbr.cz/study/courses/TIN/public/Texty/oporaTIN.pdf>
- [15] ČERNÁ, Ivana, Mojmír KŘETÍNSKÝ a Antonín KUČERA. *Formální jazyky a automaty I*. Elportál, Brno: Masarykova univerzita, 2006. ISSN 1802-128X.
- [16] Prefix, Sufix. *Teorie konečných automatů, regulárních gramatik, jazyků a výrazů* [online]. 2008 [cit. 2016-03-14]. Dostupné z: <http://iris.uhk.cz/tein/teorie/slovo.html>
- [17] Formální gramatika. *Formální gramatika - Vojtěch Hordějčuk* [online]. 2008 [cit. 2016-03-14]. Dostupné z: <http://voho.cz/wiki/formalni-gramatika/>
- [18] office.lasakovi.com [online]. [cit. 2016-04-12]. Dostupné z: <http://office.lasakovi.com/excel/vba/co-je-VBA/>
- [19] LAURENČÍK, Marek a Michal BUREŠ. *Programování v Excelu 2007 & 2010: záznam, úprava a programování maker*. 1. vyd. Praha: Grada, 2011. Průvodce (Grada). ISBN 978-80-247-3448-4.
- [20] Levenshtein distance calculator. *Convertforfree.com* [online]. [cit. 2016-04-12]. Dostupné z: <http://www.convertforfree.com/levenshtein-distance-calculator/>

## **SEZNAM PŘÍLOH**

Příloha č. 1.: Soubor obsahující slovníky a formuláře pro výpočet Leveshteinovi vzdálenosti [soubor LevenshteinDistance.xlms]